ARL-TR-8172 ● Sep 2017

US Army Research Laboratory

A Fuel-Sensitive Reduced-Order Model (ROM) for Piston Engine Scaling Analysis

by Drew Culpepper, Luis Bravo, and Simon Su

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8172 ● Sep 2017

US Army Research Laboratory

A Fuel-Sensitive Reduced-Order Model (ROM) for Piston Engine Scaling Analysis

by Drew Culpepper and Luis Bravo
Vehicle Technology Directorate, ARL

Simon Su
Computational and Informational Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED (From - To)
September 2017	Technical Report	1 June 2016–31 January 2017

4. TITLE AND SUBTITLE
A Fuel-Sensitive Reduced-Order Model (ROM) for Piston Engine Scaling Analysis

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Drew Culpepper, Luis Bravo, and Simon Su

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
US Army Research Laboratory
ATTN: RDRL-VTP
Aberdeen Proving Ground, MD 21005-5066

8. PERFORMING ORGANIZATION REPORT NUMBER
ARL-TR-8172

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The objective of this research is to establish a novel reduced-order model (ROM) for engine liquid-length scaling analysis and to assess its validity over ranges relevant to US Army vehicle-propulsion platforms. The formulation stems from the well-known Siebers' model for diesel sprays that applies heat and mass transfer principles for zero-dimensional, mixing-limited conditions. This work extends this foundational model to include 1-D transient framework with multiphysics capability. Fuel libraries have been developed including pure fuels like n-dodecane, cetane, and tetradecane and also surrogate mixtures to emulate jet-propellant (JP-8) fuel properties. Companion computational fluid dynamics simulations resolving the transient 3-D spray behavior have been performed to validate the model. Further vetting was conducted against measurements for the various cases of interest. The cases include an evaporating single-plume spray and a single-cylinder moving piston case near top dead center at diesel-engine conditions. The ROM provides a real-time engineering analytical tool for liquid-length scaling that may be used toward optimizing engine performance.

15. SUBJECT TERMS
reduced-order model, ROM, engine scaling, spray, computational fluid dynamics, fuels

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Drew Culpepper
Unclassified	Unclassified	Unclassified	UU	56	19b. TELEPHONE NUMBER (Include area code) 410-278-9719

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

Contents

List of Figures

List of Tables

1. Introduction

A central challenge engine manufactures face is meeting the power requirements with ever higher combustion efficiency. The US Army's need for faster, leaner, more powerful engines is motivating the power and energy industry to produce technologies that can meet these requirements. Novel engine designs must provide fuel flexibility, improved combustion efficiencies, and power densities to ensure battlefield dominance for air and terrestrial operations. For Army operations, reducing the battlefield fuel consumption (and payload) can have a significant impact on improving the military operational capability, enhancing force projection, and contributing to mission success.

In addressing these challenges it is important to understand the quality of the fuel–air-mixture formation process as it directly affects the combustion performance (Bravo et al. 2014b, 2015a, 2015b, 2016a; Ma et al. 2014). Mixture formation starts with the injection of the compressed liquid fuel, which mixes and vaporizes with the surrounding oxidizer leading to ignition and sustained combustion (Ray et al. 2016). It is important to understand the behavior for the liquid length in engine sprays in order to develop mixture-formation strategies and avoid adverse outcomes such as spray hitting the piston wall and causing increased emissions. To deal with this trade-off, a number of spray models and approaches have been developed. This report compares 2 such models: a reduced-order model (ROM) developed from the work of Naber and Siebers (Naber and Siebers 1996; Siebers 1999) and a full-order model (FOM) based on 3-D–computational fluid dynamics (CFD) simulations. The FOM is used as an initial benchmark, as it has several advantages including its formulation based on fluid governing equations, multidimensionality, and numerical accuracy. As usual with such models, the main disadvantage of the FOM simulation is its steep computational cost, long simulation time (order of days), and execution on high-performance-computing (HPC) platforms that may not be readily available in the field. By contrast, a ROM takes fewer inputs and can run on a standard desktop or laptop equipped with a more common programming language (FORTRAN, C++, etc.). The main disadvantage of the ROM is that it is limited to lower dimensions and steady-state conditions and introduces assumptions (e.g., mixing-limited vaporization that limits its applicability).

The original Siebers model (Siebers 1999) introduced a mixing-limited scaling law (zero-dimensional) for the steady-state calculation of an evaporating liquid spray plume. Siebers' model combined measured input functions (mainly from curve-fitting American Petroleum Institute [API] enthalpy and compressibility

data) with empirical parameters to observed liquid lengths. This method found a good fit for most points except in the low-density range where the law overestimated the liquid length. In earlier work, Siebers and Naber had introduced a penetration correlation that estimates the transient spray state as nonvaporizing. A paper by Schihl et al. (Schihl et al. 2006) later revisited and extended the model finding good agreement with experiment in the ranges of interest.* To expand the model's scope, Schihl et al. also introduced several schemes for analyzing more realistic fuel blends, all based on representing the fuel as a surrogate mixture of pure components. A series of papers by Kurvers and Luijten (Kurvers 2009; Kurvers and Luijten 2010; Luijten and Kurvers 2010) examined the effect of real-gas corrections to the scaling law. They found some improvement when accounting for real-gas effects; however, the results continued to marginally overestimate the penetration lengths at low densities. In summary, these studies suggest the assumptions behind the model are largely valid at most conditions relevant to diesel engines, although the models struggle at densities below around 14 kg/m^3.

This report describes an implementation of the Siebers steady-state scaling law along with the Naber–Siebers penetration correlation for transient effects. The cases studied include an evaporating single-plume spray and a single-cylinder moving piston case near top dead center (TDC) at diesel-engine conditions. The results are compared to laboratory measurements and to FOM simulations. The ROM is coded in MATLAB, and the FOM is modeled using the CONVERGE commercial CFD package. The FOM requires the use of a HPC platform; in this case, the Garnet system at the Army Corps of Engineers' Engineer Research and Development Center. The ROM provides a real-time engineering analytical tool for liquid-length scaling that may be used toward optimizing engine performance.

2. ROM Methodology

The ROM consists of 2 segments. The first is a steady-state solution at thermodynamic equilibrium based on Siebers' method (Siebers 1999). Siebers derived his model, which he termed a scaling law, by combining conservation of mass, momentum, and energy in a fuel spray with phase-transition thermodynamics at the liquid length. Implementing this law requires a number of other inputs. These may be either empirical or based on theoretical concepts. The model presented here largely follows the empirical approach of Siebers (1999) and explored in detail by Schihl et al. (2006). Figure 1 depicts the concept of Siebers' framework.

* It should be noted Schihl et al. (2006) contains what seems to be a typographical error in its replication of Siebers' scaling law. The numerator of its Equation 2 contains a factor of P_a where P_s should be used instead.

A second segment determines transient behavior between injection and equilibrium. It is based on a penetration correlation derived by Naber and Siebers (1996), drawn largely on the same conceptual framework as the steady-state solution but without the imposition of thermodynamic conditions. The flow chart in Fig. 2 shows a schematic of how the ROM is implemented.

2.1 Sieber's Theoretical Framework

The Siebers ROM is conceptually based on a mixing-limited conical spray, which can be visualized as a continuous series of zero-depth disks of fuel–air blend that propagate forward from the nozzle, expanding and entraining air as they go. Each disk is gradually vaporized until the time it reaches a distance L, where all fuel is vaporized. This distance is referred to as the steady-state liquid length.

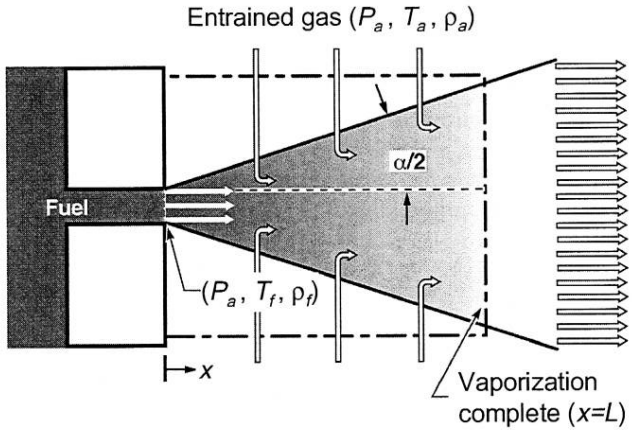

Fig. 1 Schematic of theoretical conceptual framework by Siebers (1999) in research sponsored by the US Department of Energy

The formulation begins with mass conservation

$$\dot{m}_f = \dot{m}_{f0} = \rho_f A_0 U_0 \quad , \tag{1}$$

$$\dot{m}_a = \rho_a A\, U \tag{2}$$

and momentum conservation

$$\dot{m}_{f0} U_0 = \dot{m}_f U + \dot{m}_a U \ . \tag{3}$$

Here A is the area of a disk, U its velocity, and $\dot{m}$ the mass flow rate of the specified component (f for fuel, a for entrained air). Densities are constant. Other quantities marked with a zero are values at the nozzle. All others are implicitly dependent on the distance x from the nozzle.

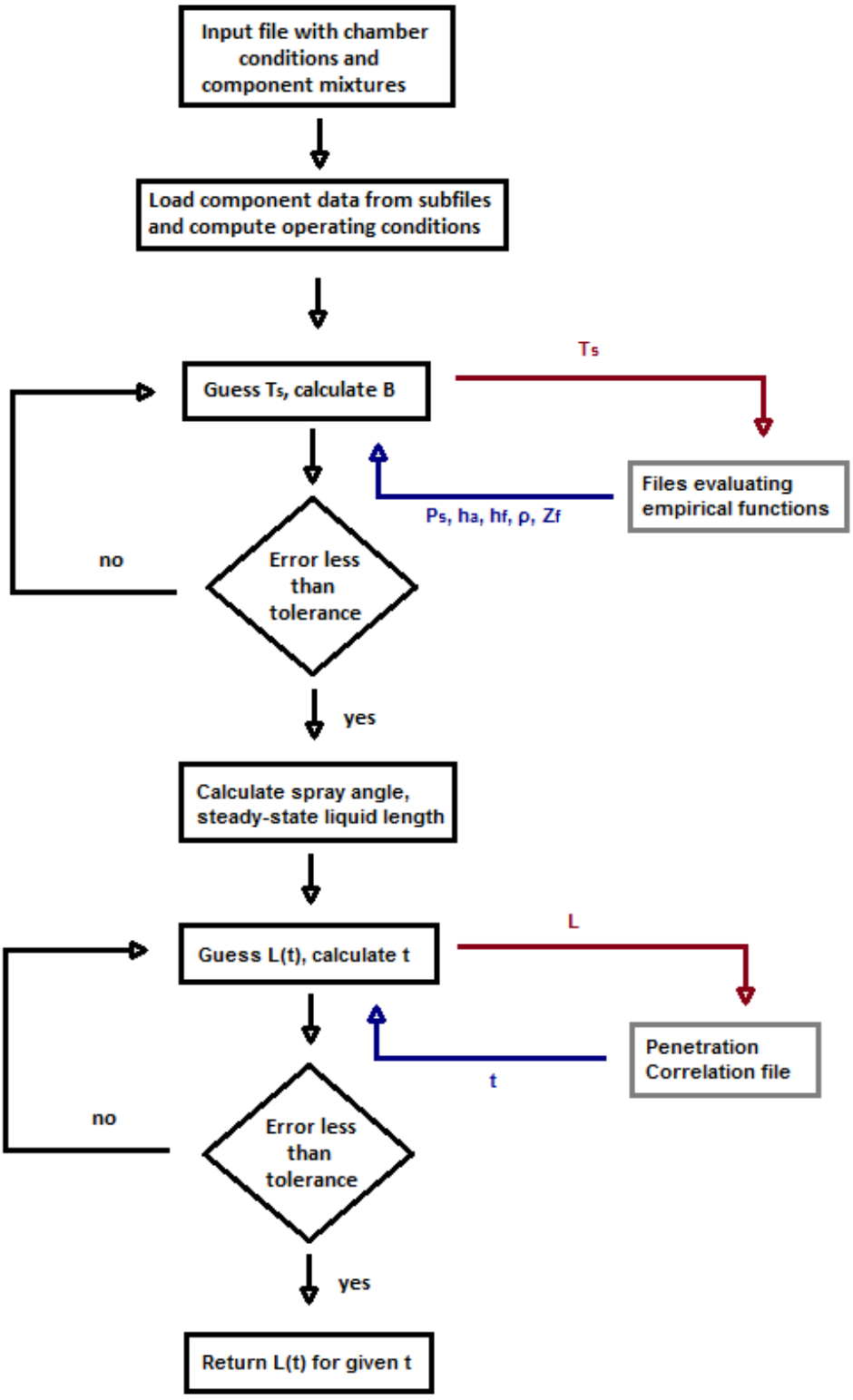

Fig. 2 Simplified schematic flow of ROM calculation

Using the conical geometry and empirical values provided by Siebers (1999), one can eliminate the velocities to solve for x

$$x = 0.62 \frac{d_{eff}}{\tan\alpha/2} \cdot \sqrt{\rho_f / \rho_a} \cdot \sqrt{\left(1 + \frac{2}{B(x)}\right)^2 - 1} \ , \tag{4}$$

with

$$B = \frac{\dot{m}_f(x)}{\dot{m}_a(x)} \ . \tag{5}$$

The liquid length L is just the x value associated with $B(L)$, but since the formula is self-referencing and too difficult to solve, this value must be determined in turn by iteration. Siebers (1999) uses thermodynamic properties of the evaporating spray to recast B in 2 further ways.

First from the enthalpy conservation between the initial final states of an air–fuel disk,

$$\dot{m}_f h_f(T_f, P_a) + \dot{m}_a h_a(T_a, P_a) = \dot{m}_f h_f(T_s) + \dot{m}_a h_a(T_s, P_a - P_s) \tag{6}$$

with specific enthalpies h_a and h_f for the air and gas respectively. Rearranging and assuming the fuel–air mix reaches thermal equilibrium (T_s, P_s) at the liquid length gives

$$B = \frac{\dot{m}_f}{\dot{m}_a} = \frac{h_a(T_a,P_a) - h_a(T_s,P_a-P_s)}{h_f(T_s) - h_f(T_f,P_a)} \ . \tag{7}$$

In this context B is known as the evaporation constant.

Second from ideal gas law

$$P = Z \cdot m \cdot MW \cdot RT$$

with compressibility Z and molecular weight MW. Since the fuel and air are both vapor at the liquid length, rearranging gives another form of B:

$$B = \frac{\dot{m}_f}{\dot{m}_a} = \frac{Z_a(T_s,P_a-P_s)}{Z_f(T_s,P_s)} \cdot \frac{P_s}{P_a-P_s} \cdot \frac{MW_f}{MW_a} \ . \tag{8}$$

Determining the evaporation constant is now equivalent to finding a T_s that satisfies

$$\frac{h_a(T_a) - h_a(T_s)}{h_f(T_s) - h_f(T_f)} = \frac{Z_a(T_s)}{Z_f(T_s)} \cdot \frac{P_s}{P_a-P_s} \cdot \frac{MW_f}{MW_a} \ . \tag{9}$$

The model is completed by input functions for the enthalpies, compressibility, and saturation pressure. The next section will detail how this evaporation condition is discovered.

2.2 Solution Method

The 2 representations of B can be reformulated as 2 functions of T_s:

$$B_h(T_s) = \frac{h_a(T_a) - h_a(T_s)}{h_f(T_s) - h_f(T_f)} \ ,$$

$$B_g(T_s) = \frac{Z_a(T_s)}{Z_f(T_s)} \cdot \frac{P_s(T_s)}{P_a - P_s(T_s)} \cdot \frac{MW_f}{MW_a} \ .$$

By comparing them, the evaporation constant can be determined for a fuel species via a root-finding algorithm that adjusts T_s until these functions are within a specified tolerance of each other:

$$\frac{B_g - B_h}{B} < \varepsilon \tag{10}$$

with the average $B = \frac{1}{2}\left(B_g + B_h\right)$ taken as the final value.

The MATLAB algorithm in this study does this by guessing a series of values for T_s and updating these guesses up or down depending on the difference found in Eq. 10. This algorithm is similar to a binary search on an array, except here the search space is the real line. As in a binary search, convergence occurs in logarithmic time, as shown in Appendix A. Appendix B has the MATLAB code for ROM.

The input values and tables of species-specific parameters are fed into files that run function files. The functions used are empirical formulas (which follow, with references). Most of these functions require further species-specific parameters, which are given for the 3 pure fuels and 4 ambient gases the ROM currently uses (see Tables 1 and 2). Values of h are in joules per gram; all others are in SI units except where specified.

Ambient enthalpies—NASA formula (McBride et al. 1993):

$$h_a(T - 298.15\ K) = \frac{R}{MW_a}\left(AT + B\frac{T^2}{2} + C\frac{T^3}{3} + D\frac{T^4}{4} + E\frac{T^5}{5} + F - G\right) \ . \tag{11}$$

Table 1 Species-specific parameters for 4 ambient gases the ROM uses

Gas	Range	A	B·10³	C·10⁷	D·10¹⁰	E·10¹⁴	F·10⁻³	G·10⁻⁴
Nitrogen	$T \leq 1000$ K	3.531	–0.1237	–5.030	24.35	–140.9	–1.047	0.000
	$T \geq 1000$ K	2.953	1.397	–4.926	0.7860	–0.4608	–0.9239	0.000
Oxygen	$T \leq 1000$ K	3.782	–2.997	98.47	–96.81	324.4	–1.064	0.000
	$T \geq 1000$ K	3.661	0.6564	–1.411	0.2058	–0.1299	–1.216	0.000
Carbon dioxide	$T \leq 1000$ K	2.357	8.985	–71.23	24.59	–14.37	–48.37	–4.733
	$T \geq 1000$ K	4.637	2.741	–9.958	1.604	–0.9161	–49.02	–4.733
Water vapor	$T \leq 1000$ K	4.199	–2.037	65.20	–54.88	177.2	–30.29	–2.908
	$T \geq 1000$ K	2.677	2.973	–7.738	0.9443	–0.4269	–29.89	–2.908

Saturation pressure—API handbook (API 1997), Formula 5A1.1*:

$$\ln P_s = A - BT_s^{-1} - C \ln T_s + DT_s^{2} + ET_s^{-2} \quad . \tag{12}$$

Table 2 Species-specific parameters for 3 pure fuels the ROM uses

Fuel	A	B	C	D·10⁶	E
Dodecane	170.27	25990	20.822	2.9759	645190
Tetradecane	130.78	20072	15.743	2.531	–1008900
Cetane	174.2	28534	21.09	2.5228	88111

Liquid density—API handbook (API 1997), Formula 6A2.13-1:

$$\frac{1}{\rho_f} = \frac{RT_c}{P_c} Z_{ra}^{\,1+(1-T_f/T_c)^{2/7}} \quad . \tag{13}$$

Liquid-fuel enthalpy—Schihl et al. (2006) curve fit of API data:

$$h_f(T,P) = \begin{cases} P_{rs} < 0.2: & AT_{rs} - B \\ P_{rs} \geq 0.2: & -CT_{rs}^{\,2} + DT_{rs} - E \end{cases} \quad . \tag{14}$$

Vaporized-fuel enthalpy—Schihl et al. (2006) curve fit of API data:

$$h_f(P_a) = A\frac{P_a}{P_c} + B \quad . \tag{15}$$

Table 3 lists Eq. 14's model coefficients for the 3 pure fuels:

Table 3 Model coefficients for Eq. 14

Fuel	A	B
Dodecane	1.277	316.3
Tetradecane	1.1718	319.2
Cetane	1.0429	320.95

* Method API 5A1.1 takes temperature input in Rankine and outputs pressure in psi absolute.

Fuel-vapor compressibility—Schihl et al. (2006) curve fit of API data:

$$Z_f = -A\left(\frac{T_s}{T_c}\right)^3 + B\left(\frac{T_s}{T_c}\right)^2 - C\left(\frac{T_s}{T_c}\right) + D \quad . \tag{16}$$

Table 4 lists Eq. 16's model coefficients for the 3 pure fuels:

Table 4 Model coefficients for Eq. 16

Fuel	A	B	C	D
Dodecane	16.85	36.104	26.425	7.5406
Tetradecane	17.924	36.143	24.71	6.6857
Cetane	16.587	34.594	24.531	6.869

Spray angle—Siebers (1999) curve fit of measurements*:

$$\tan\frac{\theta}{2} = 0.264\left[\left(\frac{\rho_a}{\rho_f}\right)^{0.19} - 0.0043\left(\frac{\rho_a}{\rho_f}\right)^{0.5}\right] . \tag{17}$$

Finally, for fuel blends the liquid length is determined through a mean evaporation constant (MEC) scheme† suggested by Schihl et al. (2006). The properties are calculated for each pure component of the blend according to Eqs. 1–7. This gives an evaporation constant B_i for each component. A weighted average gives

$$B = \sum_{i=1}^{n} x_i\, B_i \tag{18}$$

where x_i is the mass fraction of each component. This result is treated as the evaporation constant for the entire mixture and input into the scaling law to return the blend's liquid length.

2.3 Transient Modeling

The scaling-law methodology is capable only of estimating the liquid length at steady state. The Siebers framework only attempts to capture the behavior of the spray at steady state. To account for transient behavior, the ROM is modified according to the work presented by Naber and Siebers (1996). In their paper, Naber and Siebers use the same conceptual model as in this report's Section 2.1 to introduce a correlation for spray growth over time. The major new assumption of this transient model is to ignore vaporization and substitute spray length for liquid penetration. As a result, the transient model relies entirely on the kinematics of the

* Strictly speaking, the factor of 0.264 varies by nozzle, but this parameter showed no clear empirical relationship to diameter. For the nozzles studied by Siebers (1999) this value did not vary much, so we use the mean of those nozzles as a default.

† Among other schemes tested by Schihl et al. (2006), a simple weighted average of the component liquid lengths was also found to be adequate, but Schihl et al. found the MEC scheme to be slightly more accurate.

spray. The resulting transient length is therefore only approximate, but this study shows it can be suitably modified. The derivation is otherwise similar to that of steady state in Siebers (1999). Naber and Siebers (1996) first define a length scale:

$$L^+ = \frac{d_{eff}}{0.66\,\tan\alpha/2} \cdot \sqrt{\rho_f/\rho_a} \quad . \tag{19}$$

This length scale also appears implicitly in the steady-state scaling law. Time and the time-dependent liquid length can be related by reformulating them as

$$L(t) = L^+ \cdot \tilde{L}(\tilde{t}) \tag{20}$$

and

$$t = \frac{L^+}{U_f} \cdot \tilde{t} \tag{21}$$

with normalized variables $\tilde{L}$ and $\tilde{t}$. Here, U_f is the injection velocity of the fuel as determined by the Bernoulli law

$$U_f = C_v \sqrt{2 \cdot \frac{P_f - P_a}{\rho_f}} \tag{22}$$

given fuel-injection pressure P_f and an empirical-velocity coefficient C_v. The Bernoulli law applies to the fluid velocity, not the liquid-length growth, so C_v should be close to unity. Upon analysis of the validation data, we found that observed liquid lengths grew 3–5 times slower than the injection velocity. Adjusting C_v amounts to modifying the velocity of the spray, which can be done until it matches the empirical value of the liquid. Nozzles examined in Siebers' paper (1999) had velocity coefficients between 0.8 and 1 for the entire spray. The value we use for the liquid portion is 0.31, which most closely approximates the observed behavior of the liquid length.

Applying the kinematics of the spray using Eqs. 1–3, one may achieve a correlation between Eqs. 20 and 21:

$$\tilde{t} = \frac{\tilde{L}}{2} + \frac{\tilde{L}}{4}\sqrt{1 + 16\tilde{L}^2} + \frac{1}{16}\ln\left(4\tilde{L} + \sqrt{1 + 16\tilde{L}^2}\right) \quad . \tag{23}$$

This correlation is too difficult to invert, so it must be solved numerically. Our implementation closely resembles the iterative algorithm described in the previous section to determine the evaporation constant. Given a time t, a guess is made and updated for $\tilde{L}$ until Eq. 23 evaluates to a time sufficiently close to t.

As noted by Naber and Siebers (1996), the transient correlation is unbounded, so it will not accurately demonstrate steady-state behavior. This is corrected with a piecewise function in time. First, the steady-state liquid length is determined according to the Siebers scaling law. The time required to reach this distance is calculated as t_L, according to Eq. 23. For times before t_L, liquid lengths are determined according to the penetration correlation. For times after, the result from the scaling law is used.

3. FOM Methodology

The CONVERGE 3D–CFD solver, developed by Convergent Science, Inc., has been adopted in this study to perform detailed spray simulations at realistic engine-operating conditions. CONVERGE is a compressible Navier–Stokes solver. It is based on a first-order predictor–time-integration scheme and a choice of second or higher order finite-volume schemes for spatial discretization. It features a nonstaggered, collocated computation grid framework using a Rhie–Chow interpolation technique to avoid spurious oscillations. An efficient geometric multigrid treatment is used to solve the pressure equation, and a parallel computing implementation is based on implementations of either OpenMP or Message Passing Interface protocols. It provides the option of increasing resolution locally through static fixed-grid embedding and dynamically through Adaptive Mesh Refinement activated through user-specified criteria. Additionally, it uses state-of-the-art Eulerian–Lagrangian spray models to describe particle transport such as liquid-fuel injection (Bravo et al. 2016, 2017).

In this study, the gas phase is described using the Favre-Averaged Navier–Stokes equations, and the renormalization group's (RNG's) $k-\epsilon$ model for low-resolution turbulence studies has been adopted. The compressible system of transport equations for mass, momentum, energy, and species transport are presented here in a Reynolds-Averaged Navier–Stokes (RANS) framework.

3.1 Navier–Stokes Governing Equations

Conservation of Mass:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial \bar{\rho}\widetilde{u_j}}{\partial x_j} = 0 \ . \tag{24}$$

Conservation of Momentum:

$$\frac{\partial \bar{\rho}\widetilde{u_i}}{\partial t} + \frac{\partial \bar{\rho}\widetilde{u_i}\widetilde{u_j}}{\partial x_j} = \frac{\partial \bar{P}}{\partial x_i} + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial \widetilde{u_j}}{\partial x_i} + \frac{\partial \widetilde{u_j}}{\partial x_i}\right) - \frac{2}{3}\mu\frac{\partial \widetilde{u_k}}{\partial x_k}\delta_{ij}\right] + \frac{\partial}{\partial x_i}\left(-\bar{\rho}\widetilde{u_i'u_j'}\right) \ . \tag{25}$$

The last term is denoted as the Reynolds stresses of the system, $\tau_{ij} = -\bar{\rho}\widetilde{u_i'u_j'}$. It needs to be modeled to provide mathematical closure and to account for turbulence effects. Turbulent viscosity formulation in RANS typically requires the modeled Reynolds stress for the Standard $k-\epsilon$ and RNG to be represented as,

$$\tau_{ij} = \bar{\rho}\widetilde{u_i'u_j'} = 2\mu_t S_{ij} - \frac{2}{3}\delta_{ij}\left(\rho k + \mu_t \frac{\partial \widetilde{u_i}}{\partial x_i}\right) . \quad (26)$$

Hence, turbulent viscosity is defined as $\mu_t = C_\mu \rho(k^2/\epsilon)$, where k and ϵ are calculated classically by transporting 2 representative equations.

3.2 Spray CFD Models

Spray modeling is treated using the "blob" injection method of Bravo and Kweon (2014a). Blobs of a characteristic size are injected following a statistical distribution into the computational domain. Primary and secondary breakups are subsequently simulated based on the Kelvin–Helmholts (KH) and Rayleigh–Taylor (RT) instability methods. Note that the breakup length is not determined a priori (breakup-length concept) and is calculated as a part of the solution.

In the KH wave model, atomization is treated using stability analysis for liquid jets. The breakup of the injected blobs and resulting drops of radius r_o are calculated by assuming that the drop radius is proportional to the wavelength of the fastest-growing unstable surface wave Λ_{KH}.

It is written as

$$r = B_0 \Lambda_{KH} \quad (27)$$

where B_0 is a model constant. The droplet size, and its change of radius, is in the following way,

$$\frac{dr_0}{dt} = -\frac{(r_0 - r)}{\tau_{KH}} , \quad (28)$$

where the breakup-time constant, τ_{KH} , is calculated as

$$\tau_{KH} = \frac{3.726 B_1 r_0}{\Lambda_{KH}\Omega_{\mathrm{KH}}} \quad (29)$$

and the maximum growth rates Ω_{KH} and corresponding wavelengths Λ_{KH} have been simplified and defined as follows:

$$\Omega_{\mathrm{KH}}\left(\frac{\rho_l a^3}{\sigma}\right) = \frac{0.34 + 0.38 We_g^{1.5}}{(1+Z)(1+1.4T^{0.6})} \quad (30)$$

and

$$\frac{\Lambda_{KH}}{a} = 9.02 \frac{(1+0.45Z^{0.5})(1+0.4T^{0.7})}{(1+0.87We_g^{1.67})^{0.6}} \quad (31)$$

where

$Z = We_l^{0.5}/Re_l$, $T = We_g^{0.5}$, $We_l = \rho_l U^2 a/\sigma$, $We_g = \rho_g U^2 a/\sigma$, and $Re_l = Ua/\nu_l$. The present RT mechanism formulation includes viscosity variations in the growth rate equation,

$$\omega_{RT} = -k_{RT}^2 \left(\frac{\mu_l+\mu_g}{\rho_l+\rho_g}\right) \sqrt{k_{RT}\left(\frac{\rho_l-\rho_g}{\rho_l+\rho_g}\right) a - \frac{k_{RT}^3\sigma}{\rho_l+\rho_g} + k_{RT}^4 \left(\frac{\mu_l+\mu_g}{\rho_l+\rho_g}\right)^2} \quad (32)$$

where k_{RT} is the wave number, μ_l is the liquid viscosity, μ_g is the gas viscosity, ρ_l is the liquid density, ρ_g the gas density, a is the deceleration of the drop, and σ is the liquid surface tension.

4. Computational Results

In Section 4.1 the 2 models are systematically demonstrated and compared. The ROM is also compared to a transient CFD FOM described in (Senecal et al. 2007). Section 4.2 compares the ROM output against laboratory measurements using a mass threshold of 90% that would correspond to the penetration depth at which 90% of the fuel mass remains in the liquid phase. The steady state also fluctuates with time, so to compare with ROM output the steady-state measurement and FOM results are averaged time or ensemble averaged.

4.1 ROM and FOM Comparison

The FOM simulations discussed in this section are all modeled following experimental specifications for constant-volume spray vessels. To get a baseline before comparing to the FOM established in Section 3, we first compare the ROM to output from MoSES, a FOM described in (Senecal et al. 2007). Figures 3 and 4 show how the steady-state and transient parts of a ROM for n-dodecane compare against the referred study. Good agreement is obtained between the simulations and measurements, so Fig. 3 uses the validation data as a proxy for the (Senecal et al. 2007) FOM. For this case, the ROM is run for a nozzle diameter of 0.246 mm, a chamber temperature of 1000 K, and an injection pressure of 138 MPa.

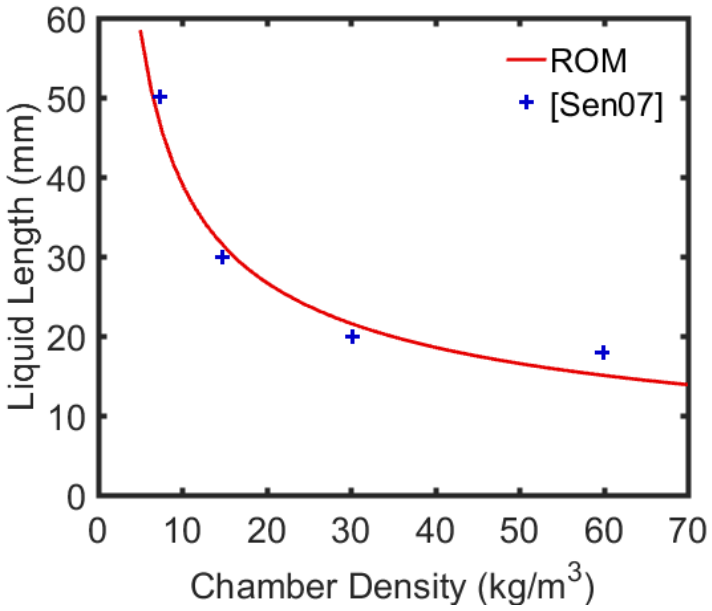

Fig. 3 Comparison of steady-state ROM with FOM described by Senecal et al. (2007); FOM output is taken at the validation data

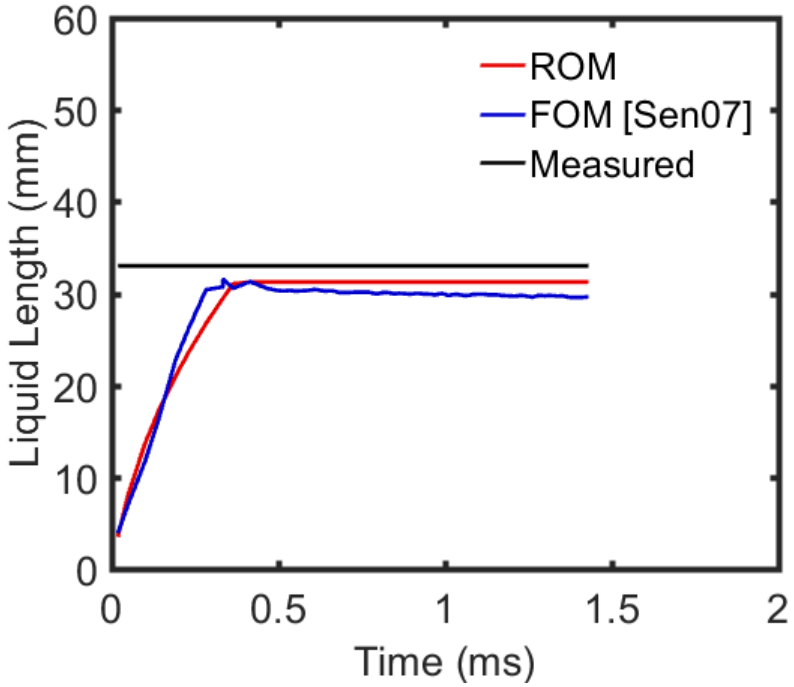

Fig. 4 Comparison of transient ROM with FOM and measurements described by Senecal et al. (2007) for chamber density of 14.8 kg/m^3; FOM output is taken at 98% mass threshold, and measured value represents a steady state of observation

For each steady state, the results are within a few percentage points of each other. Likewise, the transient behavior of both models generally agrees, though the (Senecal et al. 2007) FOM appears to converge faster than the ROM.

Moving now to the FOM presented in this study, 5 HPC simulations were executed on the HPC to compare with ROM output at steady state. For comparison to the ROM, chamber pressure was varied over the 5 cases, which were otherwise identical. The geometry used was a constant volume spray vessel. Its dimensions and conditions were as follows:

Chamber diameter	110 cm
Chamber length	110 cm
Nozzle diameter	0.15 mm
Chamber gas	Nitrogen (N_2)
Fuel	n-Dodecane
Initial ambient temperature	373 K
Initial fuel temperature	850 K

Figure 5 shows the fuel-injection process through cut planes of temperature contours for various increasing times (left to right) and decreasing chamber pressures (top to bottom). The model captures faster and longer penetration lengths with lower chamber pressures while also showing higher turbulence fluctuations at the same conditions.

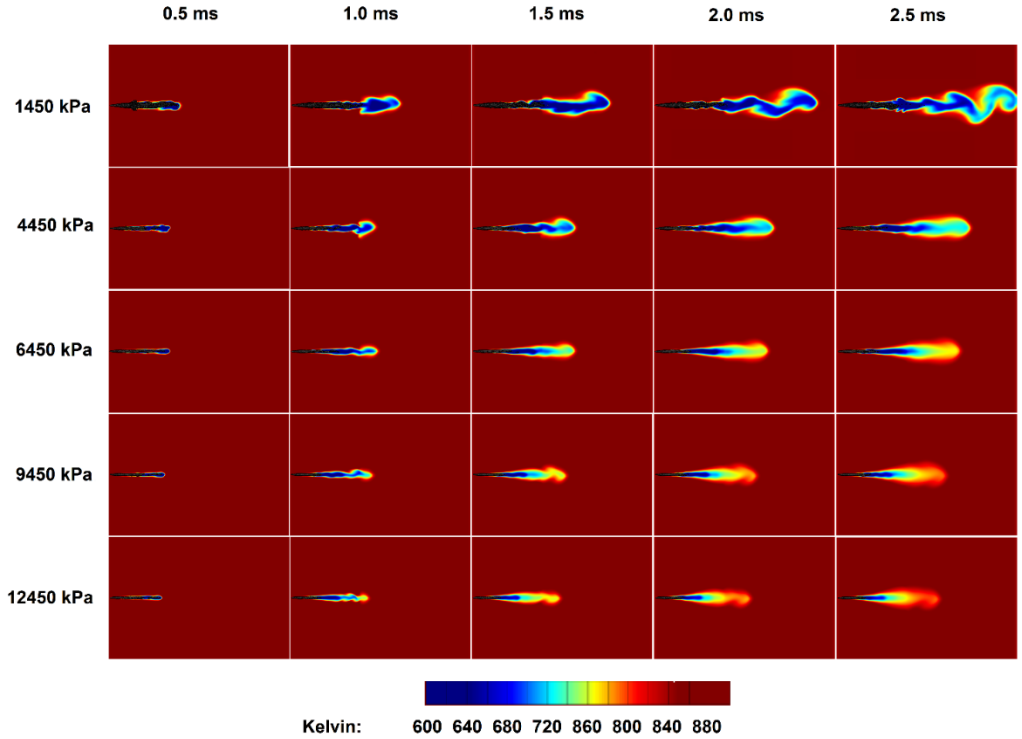

Fig. 5 Transient behavior of n-dodecane spray simulation as shown through temperature contours; liquid fuel is represented in black

Table 5 shows the liquid penetration sensitivity to fuel-mass threshold and results of 5 simulations with the previously described conditions.

Table 5 Simulated liquid lengths (mm) for different mass thresholds

Pressure (kPa)	90%	95%	97%	99%
1450	22.46	25.82	28.11	32.57
4450	15.76	17.90	19.21	21.50
6450	15.77	17.81	19.02	21.00
9450	12.95	14.42	15.30	16.76
12450	10.76	12.04	12.83	14.18

Figure 6 displays the results of the steady-state ROM against the FOM outputs at corresponding densities. The best agreement between the 2 occurs at the 90% mass threshold, except at the lower densities. The FOM also displays an unexpected dip at 17.63 kg/m^3. If this point had matched the trend of the other points, it seems likely the ROM would cross it near the 90% threshold like the higher density runs. As it stands, the ROM performs within a range of reasonable outputs from the FOM, but they do not compare as consistently at the lower densities as they do at high density.

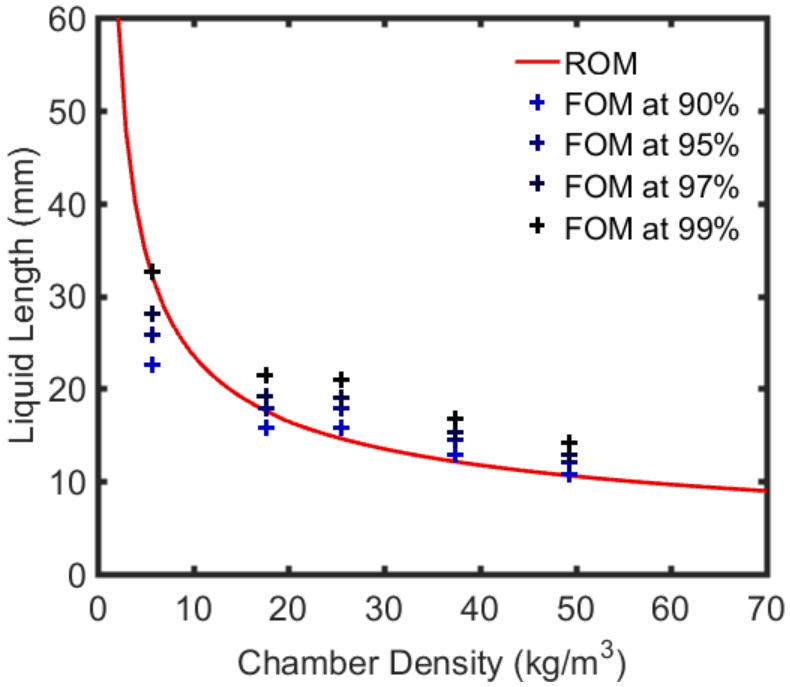

Fig. 6 Comparison of steady-state ROM with FOM at several mass thresholds

4.1.1 ROM and Experiment Comparison: Sandia National Laboratory (SNL) Experiments

Further zero-dimensional simulations were carried out to demonstrate its applicability over a wide range of conditions that have been experimentally reported. The conditions include an evaporating spray with sensitivities in the chamber operating conditions from 700 to 1300 K and across various densities at 3.6–59 kg/m^3.

Figure 7, shows the results from the single-component cetane fuel case named "ARL Model" (for the US Army Research Laboratory) and compared to the database from the Department of Energy Engine Combustion Network taken at Sandia National Laboratory. Results showed good agreement between model and experiment with less than 5% discrepancies at lower chamber densities. This can be discerned from both Fig. 7. Further, Fig. 8 demonstrates the fuel-temperature dependency is not significant for high-density cases of 30 kg/m^3.

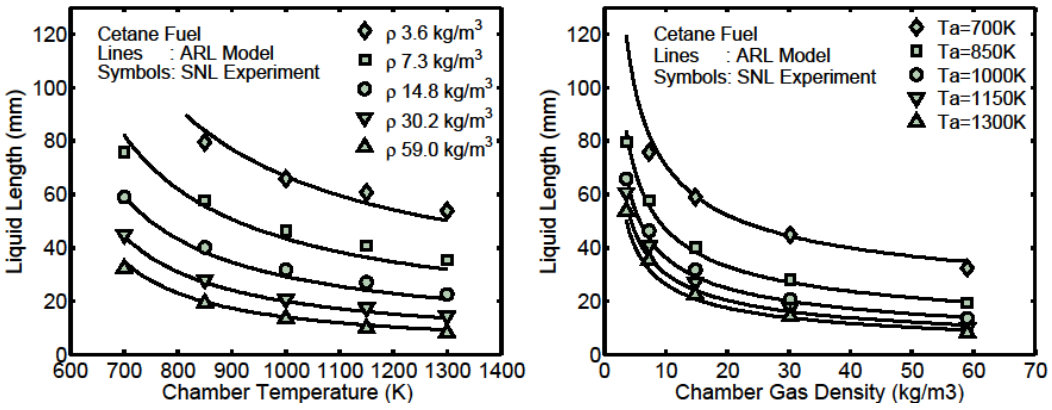

Fig. 7 Liquid-length variations via (left) chamber temperature, (right) chamber gas density space

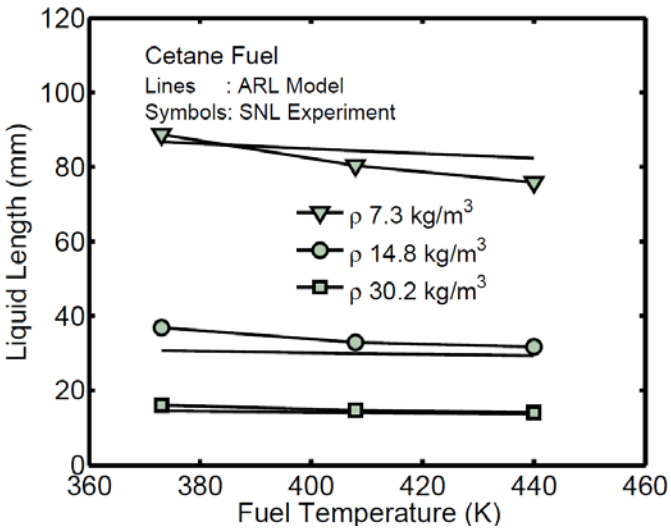

Fig. 8 Dependency of liquid length with fuel temperature

4.1.2 Impact of Fuel Properties

Similar analysis was performed with the various fuel library databases developed. In this case, we utilized the compositions for cetane, dodecane, and tetradecane to demonstrate the fuel effects on the penetration parameters. Figure 9 shows cetane provides the best prediction of the spray behavior consistently over the operating envelope. However, all fuels are able to capture the behavior and dependencies on chamber temperature. Further, we investigated the dependencies on the normalized liquid length and *B* number parameters and we were able to demonstrate a similarity law that collapses all fuels under a single curve as shown in Fig. 10.

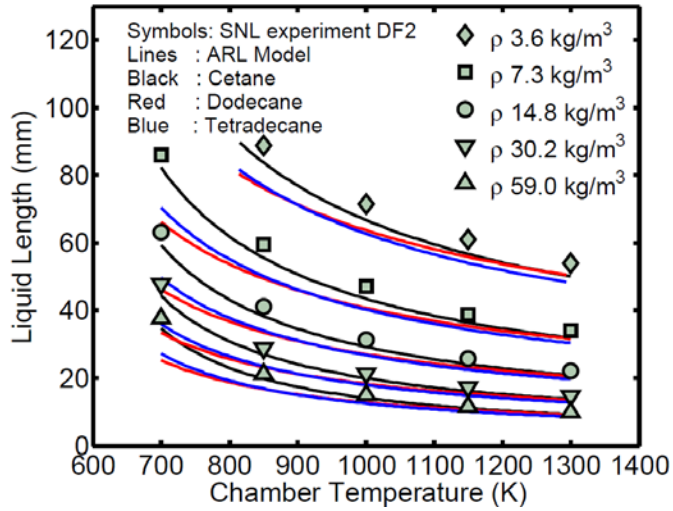

Fig. 9 Fuel effects in liquid penetration lengths over a wide operating envelope

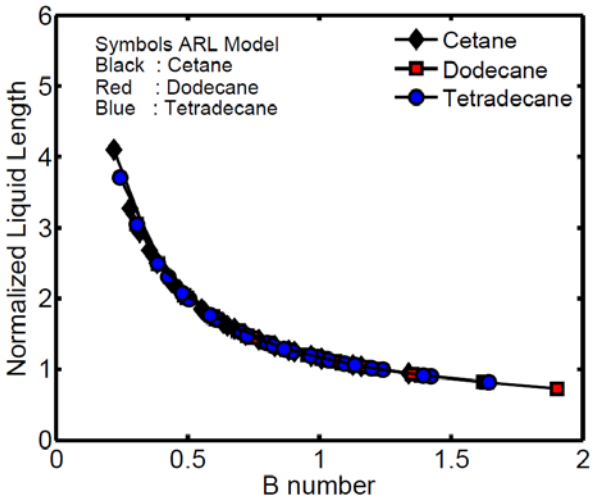

Fig. 10 Similarity in normalized liquid length and B number physical properties

4.2 ROM versus Experimental Comparison (ARL Experiments)

Figures 11 and 12 show the scaling behavior of liquid length with 3 injector nozzles for 40-, 100-, and 147-μm diameter range. Figure 11 shows the fuel effects for cetane, dodecane, and tetradecane versus chamber density for the 3 injectors. Figure 12 shows the results taking an optimized JP-8 surrogate consisting of 82% dodecane and 18% tetradecane.

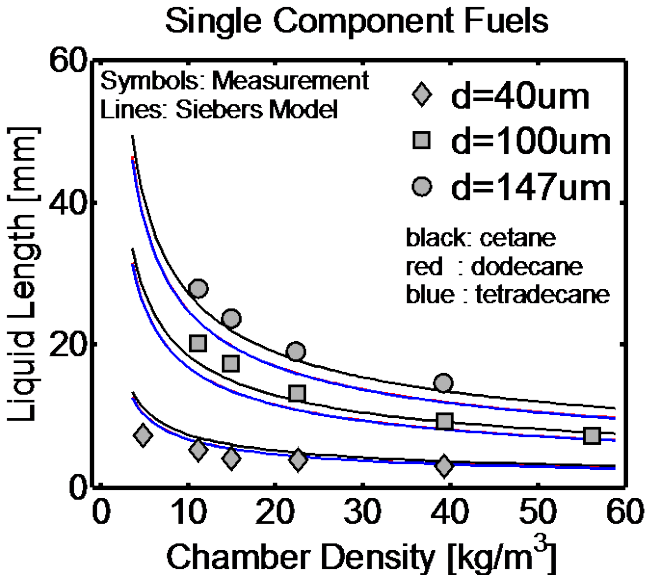

Fig. 11 Nozzle diameter and fuel effects in liquid penetration lengths over a wide operating envelope

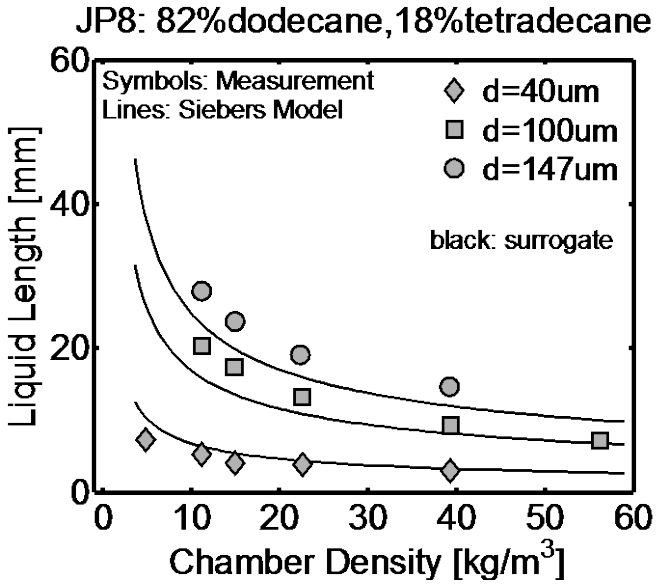

Fig. 12 Nozzle diameter with JP-8 surrogate in liquid penetration lengths over a wide operating envelope

Figure 13 shows the log-normal scale behavior between experiments and the model with good agreement for the JP-8 surrogate mixture.

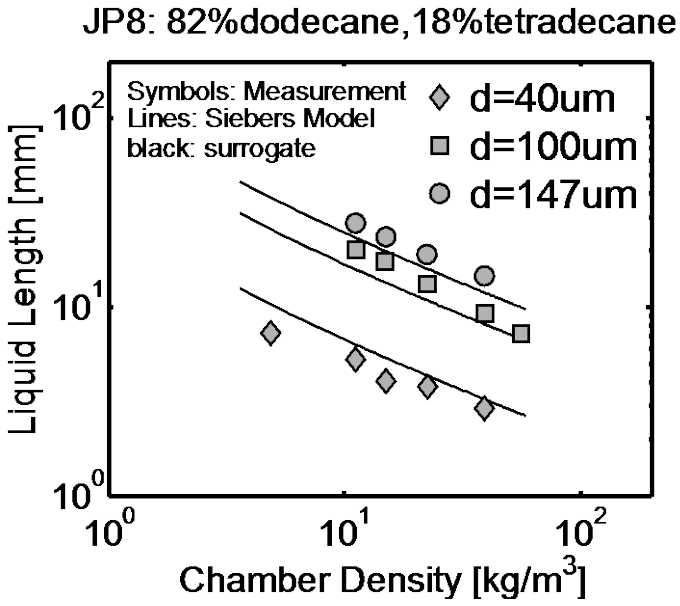

Fig. 13 Log-normal behavior of JP-8 spray

4.3 ROM versus Real Engine Measurements

The work of Espey and Dec (1995) provides a case to validate the ROM against a real diesel engine in the form of a direct-injection Cummins N-14 with a Cummins CELECT injector. Table 6 shows the basic inputs needed to compare to the ROM.

Table 6 Basic inputs for Cummins N-14 diesel engine

Nozzle Diameter	0.15 mm
Engine speed	1200 rpm
Injection pressure	68 MPa
…	…
Fuel	67.6% heptylmethylnonane
…	32.4% n-cetane

To compare with experimental data, the ROM was run for the TDC chamber conditions. An equal blend of n-dodecane, n-tetradecane, and n-cetane were used as a surrogate fuel for the mixture studied in experiment. Figure 14 shows the steady-state liquid length over 12 cycles for 14 engine configurations,* while Fig. 15 shows how average transient behavior compares.

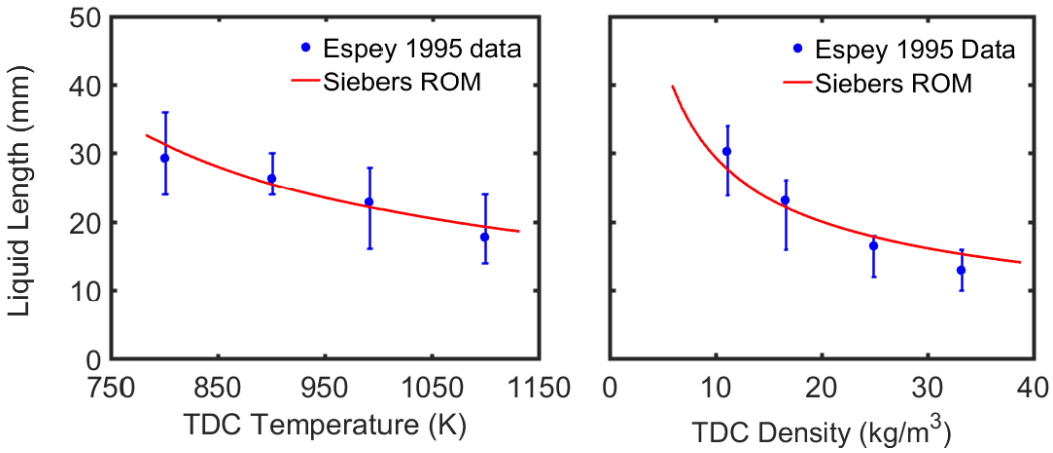

Fig. 14 ROM vs. real engine observed at steady-state: density on left is 16.6 kg/m^3, temperature on right is 992 K; error bars show observed range of values over 12 cycles, while central points mark average of all cycles and crank angles for given TDC condition

* One of the configurations (ρ_a = 16.6 kg/m^3, T_a = 992 K) is present in both data sets.

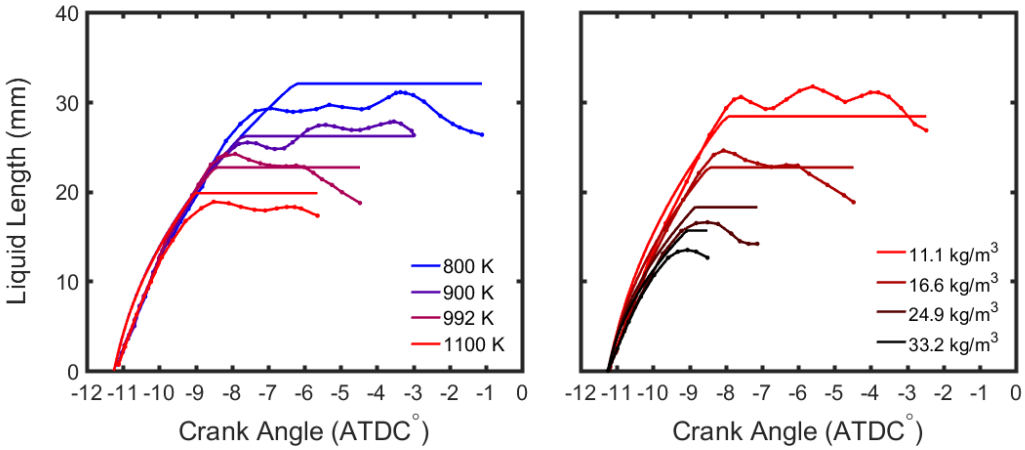

Fig. 15 ROM vs. real engine during injection: density on left is 16.6 kg/m^3, temperature on right is 992 K; solid curves are ROM output and dotted curves represent 12-cycle average of (Espey and Dec 1995) data at each crank angle

Each steady-state result from the ROM is within the range of observed values. Some discrepancy should be expected, as the ROM is based on experimental conditions measured at TDC, while liquid lengths are measured several degrees before TDC. This means the ROM is based on chamber conditions that slightly overstate the density at injection. Were the true injection conditions known the ROM results would, therefore, be somewhat lower than those calculated. Even so, the ROM results are within a few percentage points of the observed average in each case. The temperature dependence is notable because while the constant-density measurements show decreasing length with increasing temperature, they do not exhibit the smooth, upward second-derivative characteristic of the ROM. The density dependence does exhibit this behavior and largely mirrors that of the ROM, though with a slightly lower slope.

The results of Espey and Dec (1995) are also given by crank angle, allowing for a comparison of transient behavior, as given in Fig. 15. Comparing the 2 charts demonstrates a feature of the penetration correlation implemented in the ROM. Because Eq. 23 is dependent only on density and nozzle size, the ROM exhibits no transient dependence on temperature. This is consistent with the measurements, which also show temperature independence, at least until steady state. The ROM follows transient measurements reasonably, though experiment shows a more linear trend for the first moments after injection. The ROM loses its linearity much sooner, suggesting some limits to the nonvaporizing spray model.

5. Discussion

Compared to FOMs, the ROM requires relatively few inputs to run, divided into 3 categories. First are settings for the engine, such as engine speed, nozzle design, and injection parameters. The second category defines the composition of fuel and air components. The last category defines chamber-operating conditions, which include a time variable and the state variables of the fuel and air. Several other parameters are adjustable, such as the error thresholds. These, however, are more global constants that do not depend on the engine conditions being studied.

Setup for the FOM is comparatively quite complex. Beyond the need for HPC, full-dimensional CFD requires numerous input files to specify parameters such as engine geometry, spray specifications, and model numerical parameters. This allows for high customization but requires much more input from the user. If all that is needed is an estimate of liquid penetration, executing a CFD model may be excessive when a simplified model will reproduce many of the same results.

Reasonable agreement was found between the higher- and lower-order models for the conditions studied in this report. The results of the Espey and Dec (1995) measurements are also encouraging since they display good agreement even during combustion, at least until combustion products begin to obscure the spray. These results are of particular value to test the transient model, since there may be concern that a nonvaporizing spray model will not adequately capture the observed behavior. Indeed, to approximate liquid behavior, the velocity coefficient C_v must be modified to around 0.22. Preliminary agreement with measurement and full-order simulation suggests this method is a reasonable first approximation until the penetration correlation can be adjusted for vaporization.

To examine how the initial conditions affect the ROM, note that it estimates the steady-state liquid length from the product of 2 factors. The first is a normalized term

$$\tilde{L} = 0.41 \sqrt{\left(\frac{2}{B(T,\rho)} + 1\right)^2 - 1} \ . \tag{33}$$

The second is the length scale defined in Eq. 19:

$$L^+ = \frac{d_{eff}}{0.66 \tan\alpha/2} \cdot \sqrt{\rho_f / \rho_a} \ . \tag{34}$$

Other than a simple linear dependence on nozzle size, the scaling law is most affected by the temperature, pressure, and density of the chamber.* The next few figures demonstrate the effect of different chamber conditions on the ROM model outputs. Figure 16 shows the effects gas density and temperature on calculations of the B number (evaporation constant). Figure 17 shows how those values translate into a normalized length.

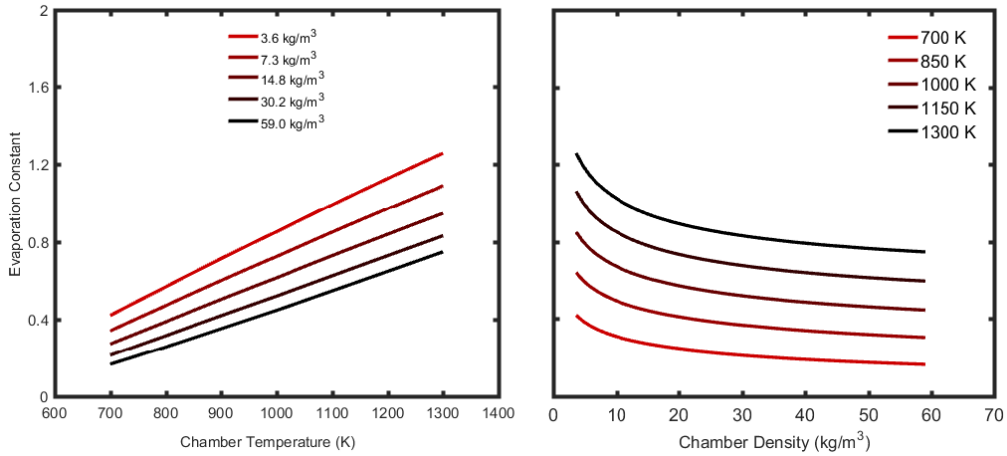

Fig. 16 Evaporation constants for cetane injected into a nitrogen chamber

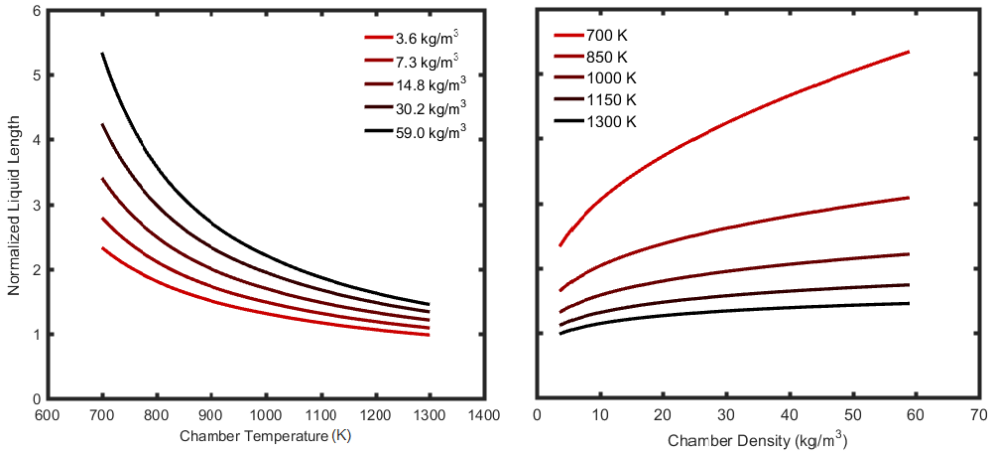

Fig. 17 Normalized liquid lengths for cetane injected into a nitrogen chamber

The fuels used will also affect the ROM's behavior. Figure 18 shows how the evaporation constant reacts to different pure fuels and also an equal blending of them. In this case the blend is almost indistinguishable from tetradecane in the evaporation constant. But, as Fig. 19 shows, when penetration is compared, the blend separates from tetradecane into a curve that is still between the fuels.

* Density is implicit in the factor of $\tan \alpha/2$, which is computed by an empirical function in Eq. 17.

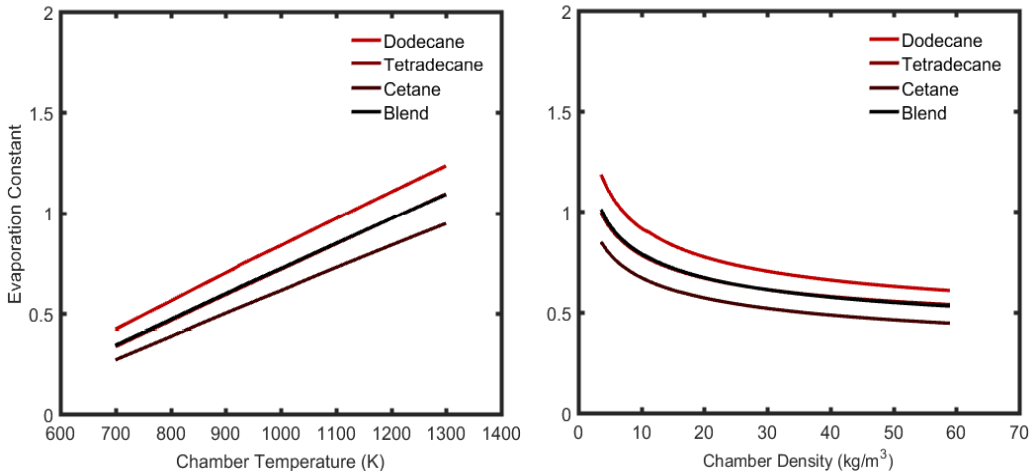

Fig. 18 Evaporation constants for 3 pure fuels and 1 equal blend; in both cases the blended evaporation constant is a simple average of that for the 3 pure fuels—and since tetradecane is close to average in each case, a significant overlap results

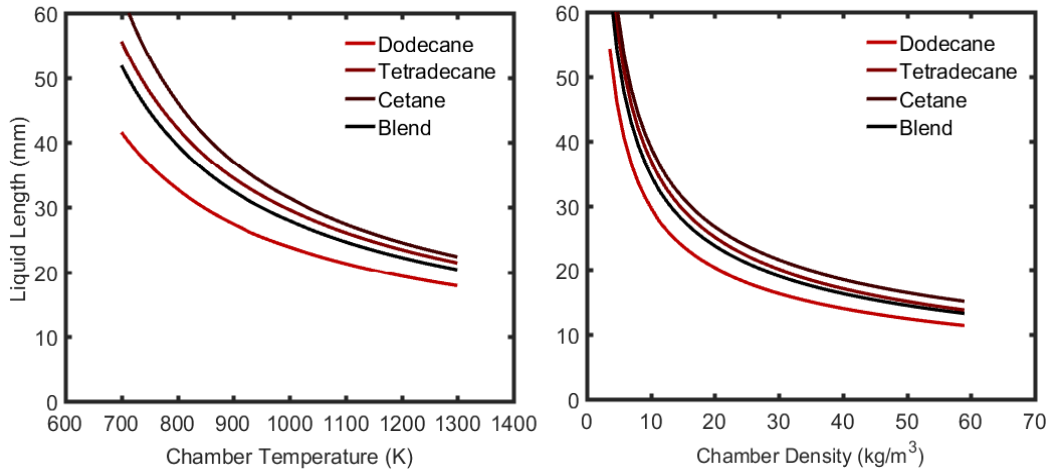

Fig. 19 Steady-state liquid lengths for 3 pure fuels and 1 equal blend

While the ROM performs comparably to the cases studied in this report, the ROM does have limitations. An engine model studied by Scarcelli et al. (2016) provides benchmark experimental measures for a gasoline direct-injection case study. Scarcelli et al. found consistent results from the same FOM described in Section 3. The FOM behaves reasonably, with liquid penetrating smoothly up to 130 mm into the cylinder on each cycle. However, it was observed through various implementations with our models, that the ROM performs poorly under the gasoline engine operating conditions (over-predicts by a factor of more than 10). The ROM would therefore not be recommended for such gasoline direct injection engines as it stands in this report. We suspect the low densities and pressures found

during gasoline injection[*] are outside the range of the empirical functions used in Section 2.2. Future work could explore this discrepancy further and modify the model should gasoline behavior be desired.

Other improvements to the ROM might include the effects of real gases. Though some nonideal effects are already captured through the use of compressibility factors, Kurvers and Luijten have shown the simple division of Eq. 8 does not adequately account for the mixture of fuel and air. The process they have developed could be used to correct the model, most prominently at higher densities.

The transient segment of the ROM could also benefit from further grounding in theory. Even with the modification to the penetration rate, the Naber–Siebers correlation is only valid for a nonvaporizing spray under the conditions noted by Naber and Siebers (1996). That the liquid fuel vaporizes is a complication that so far has been accounted for only by modifying the spray velocity. That experiment shows higher linearity than the ROM, which suggests a model with more realistic physics may provide better agreement with the data.

6. Conclusions

The objective of this research was to establish a novel ROM for engine liquid-length scaling analysis and to assess its validity over ranges relevant to Army vehicle-propulsion platforms. The formulation stems from the well-known Siebers model for diesel sprays that applies heat and mass transfer principles for zero-dimensional, mixing-limited conditions. This work extends this foundational model to include 1-D transient framework with multiphysics capability. Fuel libraries have been developed including pure fuels like n-dodecane, cetane, and tetradecane and also surrogate mixtures to emulate jet-propellant (i.e., JP-8) fuel properties. Companion CFD simulations resolving the transient 3-D spray behavior have been performed to validate the model. Further vetting was conducted against measurements for the various cases of interest. The cases include an evaporating single-plume spray and a single-cylinder moving piston case near top dead center at diesel-engine conditions. The ROM provides a real-time engineering analytical tool for liquid-length scaling that may be used toward optimizing engine performance.

In summary, this report has covered the following research:

[*] Diesel injection occurs during compression near TDC while gasoline injection occurs during expansion closer to bottom dead center. As a result, the diesel sprays studied tend to have pressures on the order of 10 bar while the gasoline spray showed typical pressures on the order of 0.1 bar, a factor of 100 lower.

- A 1-D, engine-scaling ROM was implemented based on the work of Siebers and Naber and significantly enhanced to include multifuel effects and transient injection physics.
- An FOM, based on a 3-D CFD model, for diesel fuel-injection spray was developed, implemented, and executed on the Army Engineer Research and Development Center's Garnet HPC system.
- The ROM was validated with reference data for 2 conditions: diesel fuel injection in a constant volume chamber and highly transient moving piston case.
- Model sensitivity studies were performed based on varying the combustor operating conditions (temperature, pressure, and density), nozzle fuel temperature, and fuel-injector geometry (orifice diameter) and showed accurate results.
- Suggestions have been made for future improvement such as modeling the effect of real gases at supercritical conditions, multicomponent vaporization, and more accurate thermodynamic tables for extended fuel libraries and operating envelopes (such as gasoline fuel injection).

The ROM remains largely valid in the ranges of interest, particularly in conditions common to military-operated diesel engines, though it becomes limited at the lowest densities. Despite its simplifications, it provides rapid and accurate results for liquid-spray analysis where experiment and intense CFD computations would be costly.

7. References

[API] American Petroleum Institute. Technical data book–petroleum refining. 6th ed. API: Washington (DC); 1997.

Bravo L and Kweon CB. A review on liquid spray models for diesel engine computational analysis. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2014a May. Report No.: ARL-TR-6932.

Bravo L, Ivey C, Kim S, Dokyun B. High-fidelity simulation of atomization in diesel engine sprays. Proceedings of the Center for Turbulence Research (CTR) Summer Program; 2014b Jul 28; Stanford, CA.

Bravo L, Xue Q, Som S, Powell C, Kweon CB. Fuel effects on nozzle flow and spray using fully coupled eulerian simulations. Proceedings of the ASME Power Conference; 2015a June 28–Jul 2; San Diego, CA: American Society of Mechanical Engineers. Paper No. POWER2015-49554.

Bravo L, Kim D, Tess M, Kurman M, Ham F, Kweon CB. High resolution numerical simulations of primary atomization in diesel sprays with single component reference fuels. Proceedings of the 27th Annual Conference on Liquid Atomization and Spray Systems; 2015b May 26–28, Portland, OR. Institute for Liquid Atomization and Spray Systems (ILASS-Americas).

Bravo L, Kim D, Ham H, Matusik K, Duke D, Kastengren A, Swantek A, Powell C. Numerical investigation of liquid jet breakup and droplet statistics with comparison to X-ray radiography. 52nd AIAA/SAE/ASEE Joint Propulsion Conference; 2016a July 25–27; Salt Lake City, UT.

Bravo LG, Wijeyakulasuriya S, Pomraning E, Senecal PK, Kweon CB. Large eddy simulation of high Reynolds number nonreacting and reacting JP-8 sprays in a constant pressure flow vessel with a detailed chemistry approach. J Energy Resour Tech. 2016b;183(3).

Bravo L, Ripplinger S, Samimi O. Influence of oxygen concentration on the auto-ignition and flame propagation characteristics of diesel jets with experimental comparison. Society of Automotive Engineers; 2017 Mar. SAE Technical Paper No.: 2017-01-0842.

Espey C, Dec JE. The effect of TDC temperature and density on the liquid-phase fuel penetration in a DI diesel engine. Society of Automotive Engineers; 1995 Oct. SAE Technical Paper No.: 952456.

Kurvers C. Real gas effects in mixing-limited spray vaporization models [master's thesis]. [Eindhoven, The Netherlands]: University of Technology; 2009.

Kurvers C, Luijten CCM. Real gas effects in Siebers' mixing-limited spray vaporization model. Society of Automotive Engineers; 2010 May. SAE Technical Paper No.: 2010-01-1497.

Luijten CCM, Kurvers C. Real gas effects in mixing-limited diesel spray vaporization models. Atomiz Sprays. 2010;20(7):595–609.

Ma P, Bravo L, Ihme M. Supercritical and transcritical real-fluid mixing in diesel engine applications. Proceedings of the Center for Turbulence Research (CTR) Summer Program; 2014 Jul 28; Stanford, CA.

McBride BJ, Gordon S, Reno MA. Coefficients for calculating thermodynamic and transport properties of individual species. National Aeronautics and Space Administration (US); 1993 Oct. NASA Technical Memorandum No.: 4513.

Naber JD, Siebers DL. Effects of gas density and vaporization on penetration and dispersion of diesel sprays. Society of Automotive Engineers; 1996 Feb. SAE Technical Paper No.: 960034.

Ray M, Yanga X, Konga SC, Bravo L, Kweon CB. High fidelity simulation of drop collision and vapor-liquid equilibrium of Van Der Waals fluids. Proceed Combust Inst. 2016 June.

Scarcelli R, Richards K, Pomraning E, Senecal PK, Wallner T, Sevik J. Cycle-to-cycle variations in multi-cycle engine RANS simulations. Society of Automotive Engineers; 2016 Apr. SAE Technical Paper No.: 2016-01-0593.

Schihl P, Hoogterp L, Pangilinan H. Assessment of JP-8 and DF-2 evaporation rate and cetane number differences on a military diesel engine. Society of Automotive Engineers; 2006 Apr. SAE Technical Paper No.: 2006-01-1549.

Senecal PK, Richards KJ, Pomraning E, Yang T, Dai MZ, McDavid RM, Patterson MA, Hou S, Shethaji T. A new parallel cut-cell cartesian CFD code for rapid grid generation applied to in-cylinder diesel engine simulations. Society of Automotive Engineers; 2007 Apr. SAE Technical Paper No.: 2007-01-0159.

Siebers D. Scaling liquid-phase fuel penetration in diesel sprays based on mixing-limited vaporization. Society of Automotive Engineers; 1999 Mar. SAE Technical Paper No.: 1999-01-0528.

Appendix A. Stability and Convergence Speed

As mentioned previously in the reduced-order model (ROM) methodology, the algorithm used to find the evaporation constant is based on an iterative binary search. Just as a binary search on an array requires that the array be ordered, the root-finding algorithm used here requires that B be monotonic in T_s so that the guess can unambiguously be updated based on the value it returns.

Figure A-1 gives an idea of how quickly the ROM converges given several tolerance thresholds ε computed in this equation:

$$\frac{B_g - B_h}{B} < \varepsilon .$$

Low values for ε allows for higher confidence in B, since the difference $B_g - B_h$ must be smaller to achieve it. Around the 10^{-3} tolerance level, any further gains in precision are much smaller than inconsistencies with experimental data and the FOMs, so higher precision than that is unlikely to yield more reliable results.

The method converges in logarithmic time, matching what would be expected from a binary search. The consistency of convergence time should allow for a fairly accurate estimate of the time required to run the code.*

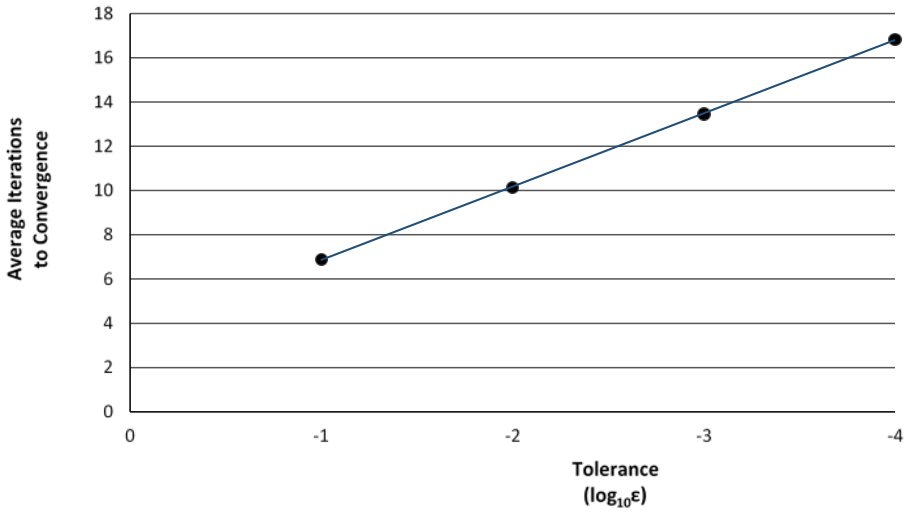

Fig. A-1 Run time to determine evaporation constant is logarithmic in the tolerance required; higher precision is shown from left to right in orders of magnitude, which requires lower tolerance

* For context, this code was timed on a Dell Latitude laptop (Intel processor at 2.5 GHz) to run about 0.5 ms per iteration. A tolerance threshold of $\varepsilon = 10^{-3}$ or $\varepsilon = 10^{-4}$ was used for most of the results in this study, translating to around 12 ms to determine B. Running the rest of the model takes on the order of a few milliseconds. Many of the condition sweeps shown in the rest of this paper were created in less than a second.

Figure A-2 shows the behavior of the evaporation constant (or B number) as a function of the saturation temperature in the ROM code. The model shows fuel sensitivities at the lower saturation temperatures up to 475 K.

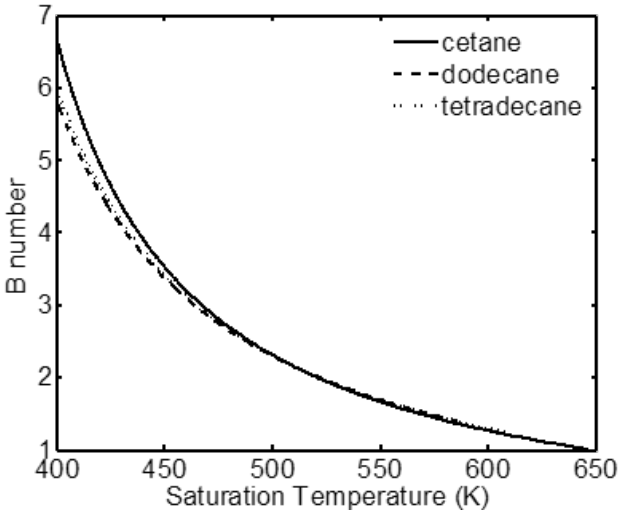

Fig. A-2 Evaporation constant dependence to saturation temperature for various single component fuels.

Figures A-3 and A-4 show the behavior of the enforced physical functions—right-hand side (RHS) and left-hand side (LHS)—used in the ROM code formulation with saturation temperature. Further, the model error calculated as the difference between the functions and fuel sensitivities is demonstrated.

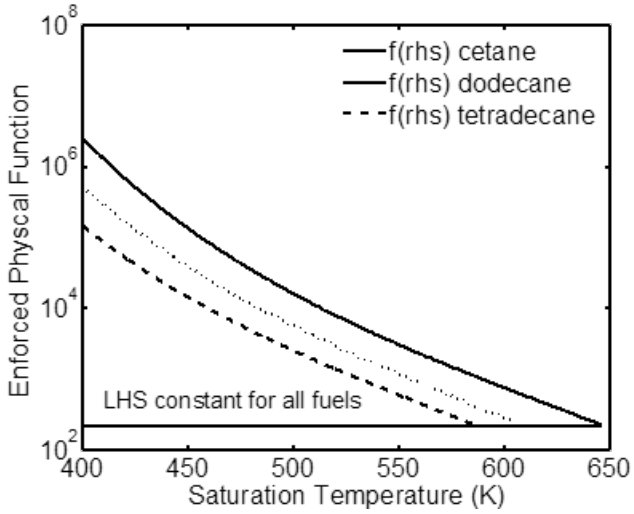

Fig. A-3 Behavior of the RHS and LHS functions with saturation temperatures of various single-component fuels.

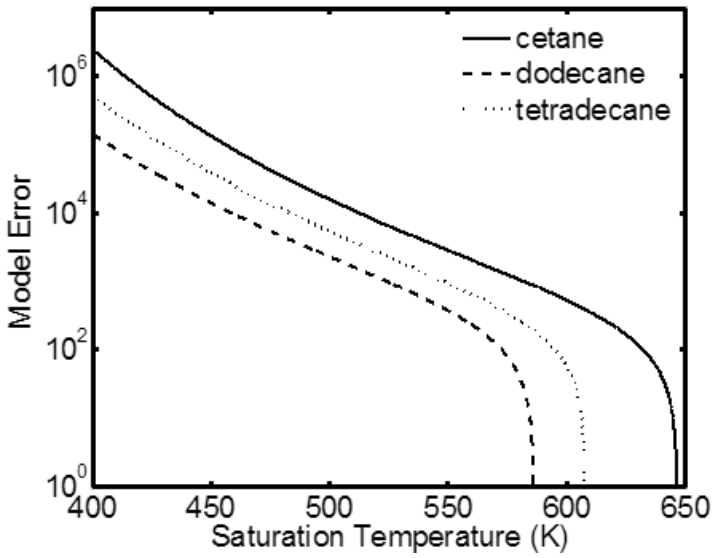

Fig. A-4 Model error (difference between RHS and LHS) with saturation temperatures of various single-component fuels

Appendix B. MATLAB Code for Reduced-Order Model (ROM)

This appendix appears in its original form, without editorial change.

The MATLAB code is executed through various modules and functions, which contain the algorithmic formulation, solution procedure, and thermodynamic properties.

```
************************************************************
***************
```

LL_ROM_RunTemplate

```
%% FLOW OF INFORMATION BETWEEN FILES

clear, clc

% Load inputs
LL_ROM_InputValues

% [MODIFICATIONS TO AMBIENT COMPONENTS & TEMPERATURE]

% Calculate ambient properties common to all fuels
AmbientProperties
AmbientMixture

% [MODIFICATIONS TO OTHER CHAMBER CONDITIONS]

% Calculate liquid length and other properties for each case
LL_ROM_BlendSolution

% [MODIFICATIONS TO TIME & ENGINE DESIGN]

% Calculate transient liquid length
LL_ROM_TransientSolution
```

```
************************************************************
***************
```

LL_ROM_InputValues

```
%% UNIVERSAL CONSTANTS
R = 8.314E-02;   % Ideal Gas Constant (bar-m3/kmol-K)
a = 0.66; % Correction factor for real spray
b = 0.41; % Correction factor for simplified scaling law assumptions

%% ENGINE SETTINGS
rpm = 1200; % Engine speed (rev/min)

% Nozzle
d = 150E-06; % Nozzle diameter (m)
NozzleAngle = 0; % Nozzle angle (degrees from cylinder axis - don't
confuse with spray angle)
Ca = 0.80; % Area-contraction coefficient (default is average of
Siebers 1999 values)
Cv = 0.22; % Velocity coefficient (default is fit to Espey & Dec
1995 data)
```

```
c = 0.2640; % Spray model parameter (default is average of Siebers
1999 values)

% Injection
Pinj = 680; % Fuel injection pressure (bar)
CAinj = -11.25; % Crank angle at fuel injection (deg ATCD)

%% FUEL & AIR COMPONENTS

% Mole fractions of fuel components:
Yf = [
    1.0, ...  % Dodecane:     fuel_flag = 1
    0.0, ...  % Tetradecane:  fuel_flag = 2
    0.0  ...  % Cetane:       fuel_flag = 3
    ];
nFuels = max(size(Yf));

% Mole fractions of ambient air components:
Ya = [
    0.7192, ...  % Nitrogen
    0.0000, ...  % Oxygen
    0.1923, ...  % Carbon Dioxide
    0.0885  ...  % Water Vapor
    ];

%% CHAMBER OPERATING CONDITIONS

t = 0.3E-03; % Time from injection (s)

% Fuel
Tf = 327; % Initial fuel temperature (K)

% Ambient air (make sure all 3 values follow gas law)
Ta = 700; % Initial ambient temperature (K)
RhoA = 0.344; % Initial ambient density (kg/m3)
Pa = 0.458; % Initial ambient pressure (bar)

%% CALCULATION OPTIONS
epsilon = 0.001; % Relative error threshold for B


**********************************************************
***************
AmbientProperties

% Sets properties for ambient air components
% From API data (Table 1C3)

AirProps = ...
  [%  MW ,   Tc ,   Pc ,    Vc ,   Zc ; ...
    28.02, 126.2, 34.00, 0.0891, 0.289; ... % Nitrogen
    32.00, 154.6, 50.43, 0.0733, 0.288; ... % Oxygen
    44.01, 304.2, 73.83, 0.0939, 0.274; ... % Carbon Dioxide
    18.02, 647.1, 220.6, 0.0559, 0.229; ... % Water Vapor
```

```
   %  [INSERT EXTRA COMPONENT VALUES AS NEEDED]
  ];

% Molecular weights
MWas = AirProps(:,1); % kg/kmol

% Values at Critical Point
Tcras  = AirProps(:,2); % Temperature (K)
Pcras  = AirProps(:,3); % Pressure (bar)
Vcras  = AirProps(:,4); % Volume (m3/kmol)
Zcras  = AirProps(:,5); % Compressibility

clear AirProps


**********************************************************
***************
```

AmbientMixture

```
%% COMBINED VALUES FOR AMBIENT MIXTURE
% Molecular Weight
MWa  = Ya*MWas; % kg/kmol

% Pseudocritical values (Modified Prausnitz-Gunn Rule)
Zcra = Ya*Zcras; % Compressibility
Tcra = Ya*Tcras; % Temperature (K)
Vcra = Ya*Vcras; % Volume (m3/kmol)
Pcra = Ya*Pcras; % Pressure (bar)
% Pcra  = Zcra * R * Tcra / Vcra; % bar

Za  = 1 ;


**********************************************************
***************
```

LL_ROM_BlendSolution

```
%% LL_ROM_BlendSolution CALCULATES THE LIQUID LENGTH OF A BLENDED
FUEL SPRAY

% Ensure Yf & Ya normalized:
if  sum(Yf)==0;  error('No  fuel  components  selected');  else  Yf  =
Yf/sum(Yf); end
if  sum(Ya)==0;  error('No  air  components  selected');  else  Ya  =
Ya/sum(Ya); end

B = 0; RhoF = 0;

% Initial Enthalpy of Chamber
Ha = AmbientEnthalpy(Ta); % kJ/kmol

Par = Pa/Pcra; % Reduced ambient pressure
```

```
global fuel_flag
for fuel_flag=1:nFuels

    if Yf(fuel_flag) == 0; continue; end % Skips fuel if component
is zero

    FuelProperties

    % Initial Enthalpy of Liquid Fuel
    Hf  = FuelEnthalpy(Par);

    %% RUN PURE FUEL SOLUTION FOR EACH FUEL

    % Search space for Ts
    TsMin = Tf; TsMax = Tcr;

    LL_ROM_PureSolution

    % Vapor Coefficient of Fuel Blend. Mean Evaporation Constant
method
    B = B + Yf(fuel_flag)*FuelB;

    % Liquid density from Rackett equation. From API data (Procedure
6A2.13)
    Tfr = Tf/Tcr;
    invRhoF = (R*Tcr) / (Pcr*MWf) * Zra^(1 + (1-Tfr)^(2/7));
    RhoF = RhoF + Yf(fuel_flag)/invRhoF; % kg/m3
end

%% CALCULATE THETA BASED ON SIEBERS CORRELATION
RhoRatio     = RhoA/RhoF;
TanHalfTheta = c * (RhoRatio^0.19 - 0.0043/sqrt(RhoRatio)) ;
HalfTheta    = atand(TanHalfTheta);

%% CALCULATE LIQUID LENGTH BASED ON SIEBERS SCALING LAW

LLnorm  = b * sqrt((1 + 2/B)^2 - 1);

LLscale = 1/a * d*sqrt(Ca) / ...
    (sqrt(RhoRatio) * TanHalfTheta);

% Liquid Length as measured vertically down axis of chamber
LL      = LLscale * LLnorm * cosd(NozzleAngle); % m
LL_mm   = LL*1000;          % mm

%% CALCULATE TIME SCALES FOR TRANSIENT SOLUTIONS

% Engine speed
AngVel = rpm * 360 / 60; % Degrees/s

% Fuel velocity
v2 = 2*(Pinj - Pa)/RhoF * 100000; % m2/s2
```

```
FuelVel = Cv * sqrt(v2); % m/s

tscale = LLscale/FuelVel;

% Time when LL(t) = LL
LLtnorm = PenetrationTime(LLnorm); % Normalized (from injection)
% Crank angle when LL(t) = LL
CAeqb = CAinj + AngVel*tscale*LLtnorm; % Degrees


**********************************************************
***************
```

LL_ROM_PureSolution

```
%% LL_ROM_PureSolution CALCULATES Ts OF A PURE FUEL ITERATIVELY
WITH A BINARY SEARCH ALGORITHM

Ts      = (TsMax + TsMin) / 2; % Initial guess for Ts
TsRange = (TsMax - TsMin) / 2; % Initial size of Ts search space
iter = 0;

accepted = false; failed = false;

while ~accepted
    %% START OF LOOP
    iter = iter + 1;
    if iter>30
        failed = true;
        continue
    end
    % Vapor pressure of fuel at aturation
    Ps = FuelSaturationPressure(Ts); % bar

    Prs = Ps/Pcr;
    Trs = Ts/Tcr;

    % Fuel enthalpy at saturation
    Hfs = SaturatedFuelEnthalpy(Prs,Trs); % kJ/kg

    % Ambient enthalpy at saturation
    Has = 100*Ya*R*AmbientEnthalpies(Ts)/MWa;

    % Fuel Compressiblity
    Zf  = FuelCompressibility(Trs);

    % Evaluate Vapor Coefficient for Iterative Solution (Ts)

    Bg    = (Za * Ps * MWf) / (Zf * (Pa-Ps) * MWa); % B value from
gas law      (Increases with Ts)
    Bh    = (Ha - Has) / (Hfs - Hf)               ; % B value from
enthalpies   (Decreases with Ts)
    FuelB = (Bg + Bh) / 2;                            % Mean of B
values adopted
```

```
    % Calculate error
    Error = (Bg - Bh) / FuelB;

    % Adjust Ts for Error value
    if abs(Error) < epsilon
        accepted = true;
    else
        TsRange = TsRange/2;
        if Error > 0
            Ts = Ts - TsRange;
        else
            Ts = Ts + TsRange;
        end
    end
end
```

```
**********************************************************
***************
```

LL_ROM_TransientSolution

```
%% LL_ROM_TransientSolution CALCULATES THE LIQUID LENGTH AT A GIVEN
TIME

% Time from injection
tnorm = t/tscale; % Normalized

if tnorm<=0
    xnorm = 0;
elseif tnorm < LLtnorm
    % Initial guesses:
    xdomain = LLnorm;
    xnorm = xdomain / 2;

    accepted = false;
    while ~accepted

        tguess = PenetrationTime(xnorm);
        Error = 1 - tguess/tnorm;

        if abs(Error)<0.001
            accepted = true;
        else
            xdomain = xdomain/2;
            if Error < 0
                xnorm = xnorm - xdomain/2;
            else
                xnorm = xnorm + xdomain/2;
            end
        end

    end
```

```
else
    xnorm = LLnorm;
end

% Transient liquid length
LLt = LLscale * xnorm; % m
```

```
************************************************************
***************
```

FuelProperties

```
%% SET PURE FUEL PROPERTIES
% From API data (Table 1C1 & Table 6A2.14)

global fuel

switch fuel_flag
    case 1
        fuel = 'Dodecane';
        MWf = 170.34;   % Molecular Weight (kg/kmol)
        Tcr  = 658;     % Critical Temperature (K)
        Pcr  = 18.2;    % Critical Pressure (bar)
        Zra = 0.2471;   % Rackett Parameter
    case 2
        fuel = 'Tetradecane';
        MWf = 198.39;   % Molecular Weight (kg/kmol)
        Tcr  = 693;     % Critical Temperature (K)
        Pcr  = 15.7;    % Critical Pressure (bar)
        Zra = 0.2270;   % Rackett Parameter
    case 3
        fuel = 'Cetane';
        MWf = 226.45;   % Molecular Weight (kg/kmol)
        Tcr  = 723;     % Critical Temperature (K)
        Pcr  = 14.0;    % Critical Pressure (bar)
        Zra = 0.2386;   % Rackett Parameter
%   [INSERT EXTRA COMPONENT VALUES AS NEEDED]
    otherwise
        error('Not a valid Fuel')
end
```

```
************************************************************
***************
```

SetHaVals

```
% From NASA data (McBride et al 1993, Table 2)
% Valid between 200K - 6000K unless otherwise specified

global HaValsHiT; global HaValsLoT; global HaValsZero;

HaValsHiT = [ ...
```

```
    %      R2C1              R2C2              R2C3              R2C4
R2C5            R3C1
     2.95257626E+00   1.39690057E-03 -4.92631691E-07   7.86010367E-
11 -4.60755321E-15 -9.23948645E+02; ... Nitrogen
     3.66096083E+00   6.56368523E-04 -1.41149485E-07   2.05797658E-
11 -1.29913248E-15 -1.21597725E+03; ... Oxygen
     4.63659493E+00   2.74131991E-03 -9.95828531E-07   1.60373011E-
10 -9.16103468E-15 -4.90249341E+04; ... Carbon Dioxide
     2.67703787E+00   2.97318329E-03 -7.73769690E-07   9.44336689E-
11 -4.26900959E-15 -2.98858938E+04; ... Water Vapor
    ];

HaValsLoT = [ ...
    %      R3C3              R3C4              R3C5              R4C1
R4C2            R4C3
     3.53100528E+00 -1.23660987E-04 -5.02999437E-07   2.43530612E-
09 -1.40881235E-12 -1.04697628E+03; ... Nitrogen
     3.78245636E+00 -2.99673415E-03   9.84730200E-06 -9.68129508E-
09   3.24372836E-12 -1.06394356E+03; ... Oxygen
     2.35677352E+00   8.98459677E-03 -7.12356269E-06   2.45919022E-
09 -1.43699548E-13 -4.83719697E+04; ... Carbon Dioxide
     4.19864056E+00 -2.03643410E-03   6.52040211E-06 -5.48797062E-
09   1.77197817E-12 -3.02937267E+04; ... Water Vapor
    ];

HaValsZero = [ ...
    %      R4C5
     0.00000000E+00; ... Nitrogen
     0.00000000E+00; ... Oxygen
    -4.73281047E+04; ... Carbon Dioxide
    -2.90848168E+04; ... Water Vapor
    ];


**********************************************************
***************
```

AmbientEnthalpies

```
% Enthalpy of ambient air, up to factor of R
% NASA data & formula (McBride et al 1993)

function Har = AmbientEnthalpies(T)

global HaValsHiT; global HaValsLoT; global HaValsZero;

% Set coefficients
if T>=1000
    HaVals = HaValsHiT; % High-temperature coefficients
else
    HaVals = HaValsLoT; % Low-temperature coefficients
end

% Set indeterminates
vars = [T; T^2/2; T^3/3; T^4/4; T^5/5; 1];
```

```
% Array of temperature components for each species
Har = HaVals*vars - HaValsZero; % kJ/kmol-R

end


************************************************************
***************
```

FuelEnthalpy

```
% Initial enthalpy of liquid fuel
% Curve fit by Schihl et al from API data

function Hf = FuelEnthalpy(Par)

global fuel
switch fuel
    case 'Dodecane'
        C1 = 1.277;
        C2 = 316.3;
    case 'Tetradecane'
        C1 = 1.1718;
        C2 = 319.2;
    case 'Cetane'
        C1 = 1.0429;
        C2 = 320.95;
end

% calculate...
Hf = C1*Par + C2;

end


************************************************************
***************
```

FuelSaturationPressure

```
% Vapor pressure at saturation temperature
% From API data (Procedure 5A1.1, Table 5A1.2)

function Ps = FuelSaturationPressure(Ts)

global fuel
switch fuel
    case 'Dodecane'
        A = 170.24;
        B = 25990;
        C = 20.822;
        D = 2.9759E-06;
        E = 645190;
    case 'Tetradecane'
```

```
        A = 130.78;
        B = 20072;
        C = 15.743;
        D = 2.3531E-06;
        E = -1008900;
    case 'Cetane'
        A = 174.20;
        B = 28534;
        C = 21.090;
        D = 2.5228E-06;
        E = 88111;
end

TsR = 1.8 * Ts; % Temperature (Rankine)

Ps = exp(           ...
      A             ...
    - B/TsR         ...
    - C*log(TsR)    ...
    + D*TsR^2       ...
    + E/TsR^2       ...
    ); % psi

Ps = 0.06895 * Ps; % bar

end
```

****

SaturatedFuelEnthalpy

```
% Curve fit by Schihl et al from API data

function Hfs = SaturatedFuelEnthalpy(Prs, Trs)

global fuel
switch fuel
    case 'Dodecane'
        A = 1562.5;
        B = 444.84;
        C = 1921.9;
        D = 4996.7;
        E = 1978;
    case 'Tetradecane'
        A = 1683.8;
        B = 469.88;
        C = 3712.2;
        D = 8085.6;
        E = 3229.2;
    case 'Cetane'
        A = 1869.7;
        B = 550.15;
        C = 929.51;
```

```
        D = 3520.3;
        E = 1283.4;
end


if Prs<0.2
    Hfs = A*Trs - B;            % kJ/kmol
else
    Hfs = -C*Trs^2 + D*Trs - E; % kJ/kmol
end

end
```

```
************************************************************
***************
```

FuelCompressibility

```
% Curve fit by Schihl et al from API data

function Zf = FuelCompressibility(Trs)

global fuel
switch fuel
    case 'Dodecane'
        A = 16.85;
        B = 36.104;
        C = 26.425;
        D =  7.5406;
    case 'Tetradecane'
        A = 17.924;
        B = 36.143;
        C = 24.71;
        D =  6.6857;
    case 'Cetane'
        A = 16.587;
        B = 34.594;
        C = 24.531;
        D =  6.869;
end

% calculate...
Zf = -A*Trs^3 + B*Trs^2 - C*Trs + D;

end
```

```
************************************************************
***************
```

PenetrationTime

```
% Normalized time required for liquid length to reach normalized
distance argument
```

```
% From Naber & Siebers

function tnorm = PenetrationTime(xnorm)

tnorm =                                                 ...
      xnorm/2                                           ...
    + xnorm/4 * sqrt(1 + 16*xnorm^2)                    ...
    + log(4*xnorm + sqrt(1 + 16*xnorm^2))/16;

end
```

INTENTIONALLY LEFT BLANK.

List of Symbols, Abbreviations, and Acronyms

3-D	3-dimensional
API	American Petroleum Institute
ARL	US Army Research Laboratory
CFD	computational fluid dynamics
FOM	full-order model
HPC	high-performance computing
LHS	left-hand side
KH	Kelvin–Helmholts
MEC	mean evaporation constant
NASA	National Aeronautics and Space Administration
RANS	Reynolds-Averaged Navier–Stokes
RHS	right-hand side
RNG	renormalization group
ROM	reduced-order model
RT	Rayleigh–Taylor
SNL	Sandia National Laboratory
TDC	top dead center

1 DEFENSE TECHNICAL
(PDF) INFORMATION CTR
DTIC OCA

2 DIR ARL
(PDF) RDRL DCM
IMAL HRA RECORDS MGMT
RDRL IRB
TECH LIB

1 GOVT PRINTG OFC
(PDF) A MALHOTRA

1 DIR USARL
(PDF) RDRL VTP
D CULPEPPER